SAND2004-1505C

Calibration, Validation, and Sensitivity Analysis: What's What and Who Cares?

Timothy Trucano and Laura Swiler

Optimization and Uncertainty Estimation Department, Org 9211
Sandia National Laboratories
Albuquerque, NM 87185

4th International Conference On Sensitivity Analysis of Model Output
March 8-11, 2004
Santa Fe, New Mexico

Phone: 844-8812, FAX: 844-0918
Email: tgtruca@sandia.gov

Outline of talk.

- Terms – in English
- Terms – in Jargon
- Terms – a tiny bit of tech-talk and mention of sensitivity analysis
- Conclusions

Our perspective in this talk:

From the day-to-day challenges of the Sandia ASCI V&V program.

As users – not providers – of sensitivity analysis.

What the Oxford English Dictionary has to say about this:

- Benchmark – "A surveyor's mark cut in some durable material…to indicate the starting, closing, or any suitable intermediate, point in a line of levels for the determination of altitudes…"
- "Benchmarking" – not a word!
- Calibration – "The act of calibrating." Calibrate – "To graduate a gauge of any kind with allowance for its irregularities."
- Prediction – "The action of predicting or foretelling future events." "A statement made before hand."
- Validation – "The action of making valid." Valid – "Possessing legal authority or force." "Of arguments, proofs, assertions, etc. Well founded and fully applicable to the particular matter or circumstances." "Of things: strong, powerful."
- Verification – "Formal assertion of the truth." "Demonstration of truth or correctness by facts or circumstances."

Additional lingo:

- Code – the software that implements the solution algorithms for a set of partial differential equations. In fact –
  - "high-performance, full-system, high-fidelity-physics predictive codes…"
- Model – I will not use this term. [But one meaning is a particular choice of input information that produces a specific output.]
- Infrastructure – the additional machinery required to run a code and produce results.
  - Meshing tools
  - Graphics tools
  - Uncertainty quantification tools
  - Other

Jargon (one specific form):

- Benchmark – a choice of information for purposes of performing calibration, verification or validation. This information is believed to be "true" for this purpose.
  - The act of Prediction does not require a benchmark.
  - Assessing the quality of a Prediction after the fact does not require benchmarks.
  - Measuring our belief in the accuracy of a Prediction does require benchmarks.
- Calibration – the process of improving the agreement of a code calculation or calculations with respect to a chosen set of benchmarks through the adjustment of parameters implemented in the code.
  - "Parameters" needs to be clarified.

Jargon Continued:

- Verification – the process of determining that requirements for the intended application are implemented correctly in the code.
- Validation – The process of determining that requirements implemented in the code are correct for the intended application.
- Computational science centers on solving partial differential equations using computer codes (for this talk, anyway):
  - Requirements are therefore centered on “correct” physics and “correct” numerical solutions.

Thus, for computational science V&V:

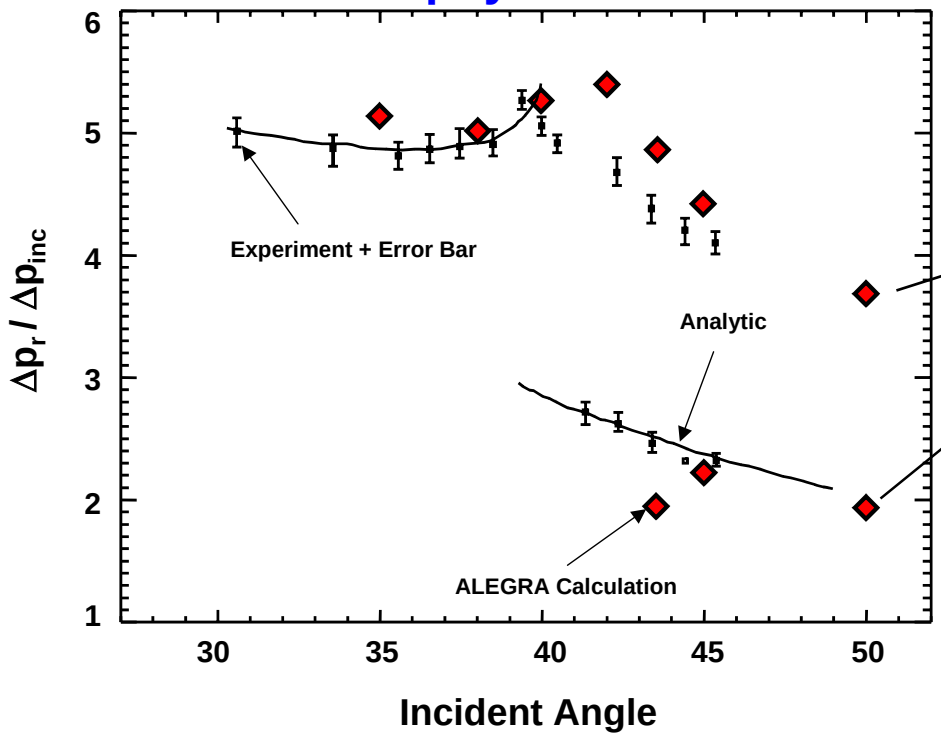
- Verification – the process of determining that equations are solved “correctly” [Roache]
- Validation – the process of determining that equations are correct [Roache]

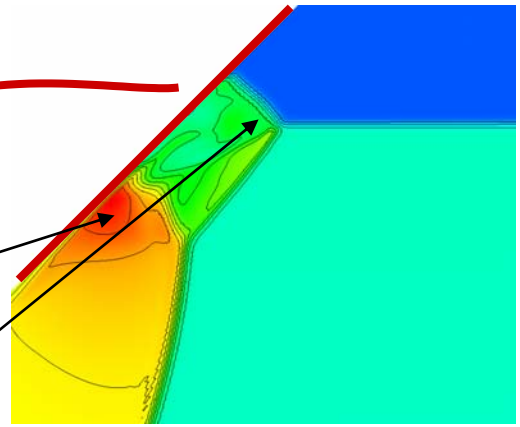
Jargon cont.:

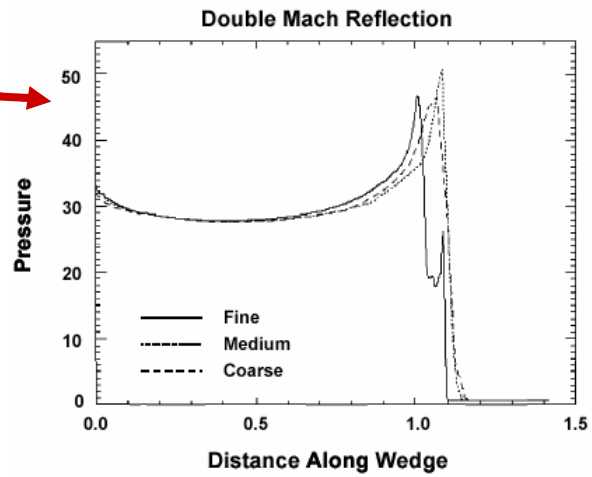
- It is generally believed that validation is a harder problem than verification because of associated philosophical problems, as well as practical problems.
  - Recall that one of Hilbert's problems is to "axiomatize" physics, which might offer a route to proving correctness. This problem remains unsolved.
  - This does not mean that verification is easy.
- Prediction – The process of performing code calculations and applying the results for code input regimes that interpolate or extrapolate the V&V benchmark domain.
  - (Note that we have been careful to distinguish V&V benchmarks, as opposed to calibration benchmarks.)

Here is calibration, verification and validation:

This is physics.







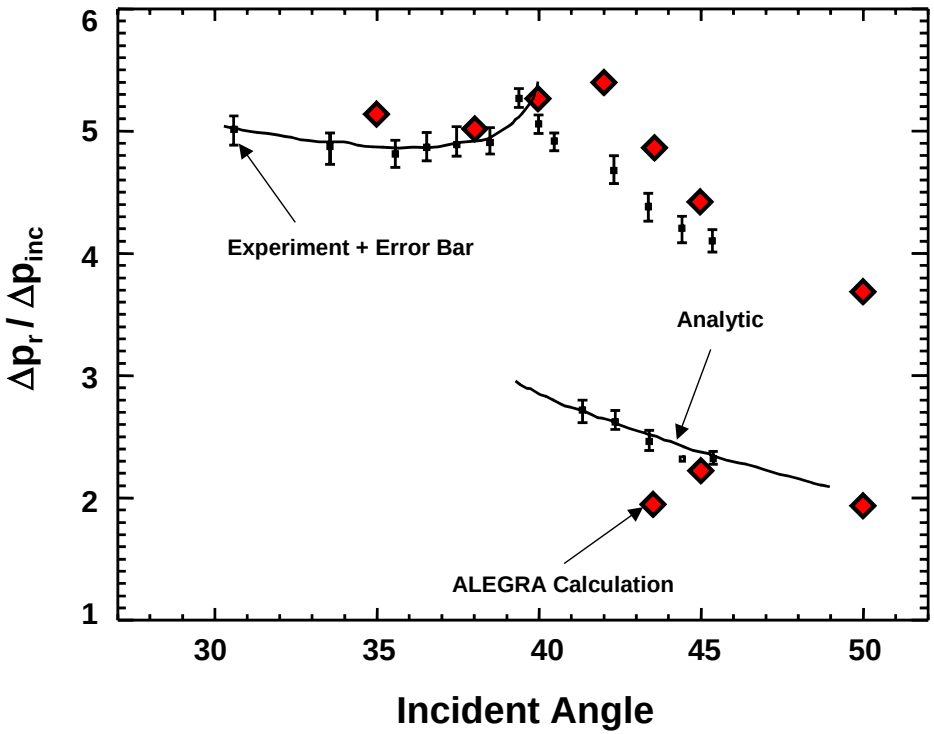
Improving agreement through calibration is both math and physics.

This is math.

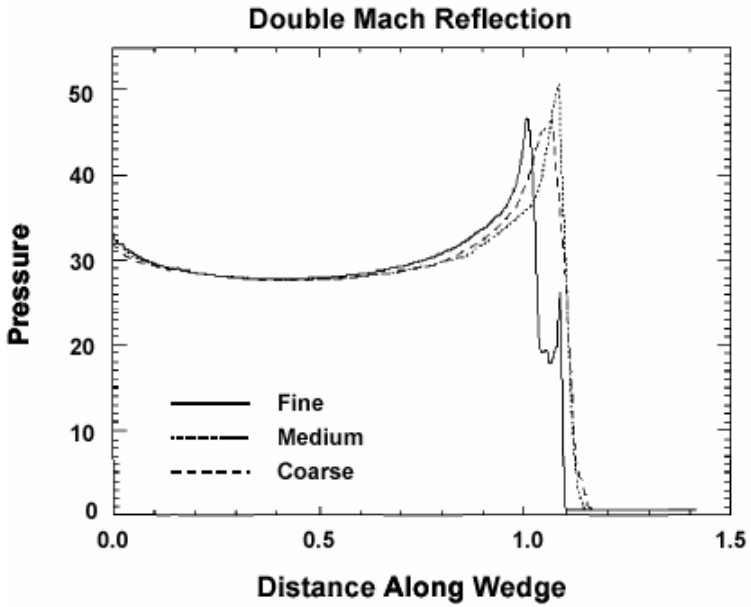
Sandia National Laboratories

Verification: Are code bugs or numerical inaccuracies corrupting the comparison with experimental data?

What is the numerical error?



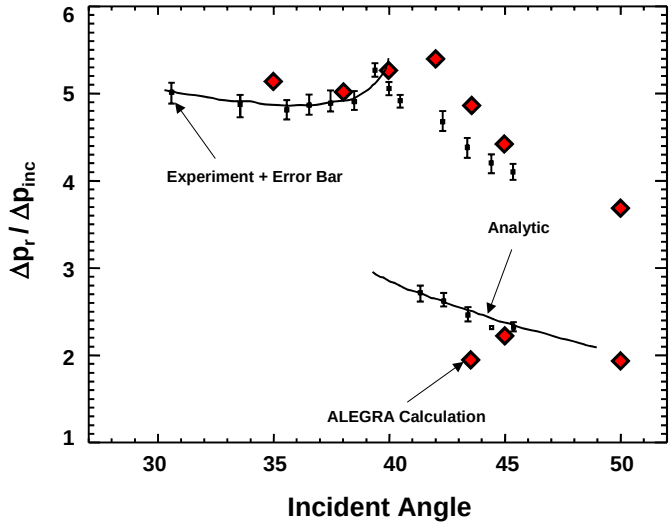
Does the code converge to the correct solution for this problem?

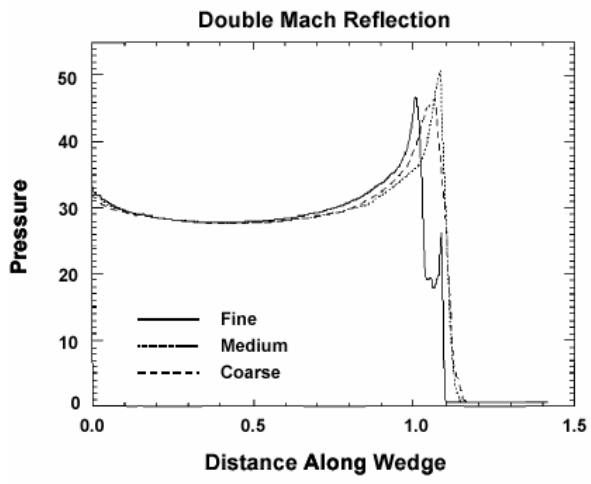


Note that choosing the mesh to better agree with experimental data is calibration, not verification (or validation).

Sandia National Laboratories

Validation: Are we performing the right calculations to compare with the right experiments in the right way to draw the right conclusions?





- **Error bars mean what?**
- **What is the numerical accuracy of the code?**
- **Is the comparison good, bad, or indifferent? In what context?**
- **Why did we choose this means to compare the data and the calculation? Is there something better?**
- **Why did we choose this problem to begin with?**
- **What does the work rest on (such as previous knowledge)?**
- **Where is the work going (e.g. what next)?**

The rub:

- Validation requires verification: computational errors in validation comparisons must be smaller than physical errors (experimental and physics in the code) to make these comparisons meaningful in the context of validation.
- Ask your favorite computational modelers what the numerical errors are in their calculations.
  - By the way, ask them to prove their answer. (After all, it IS a mathematics problem!!)
- For complex calculations, nobody really knows the answer to this question, and surely can prove little of what they know.
- The rub:

  Validation has uncertainty, both variability and lack of knowledge, deeply embedded in it.
- Understanding calibration requires performing validation.

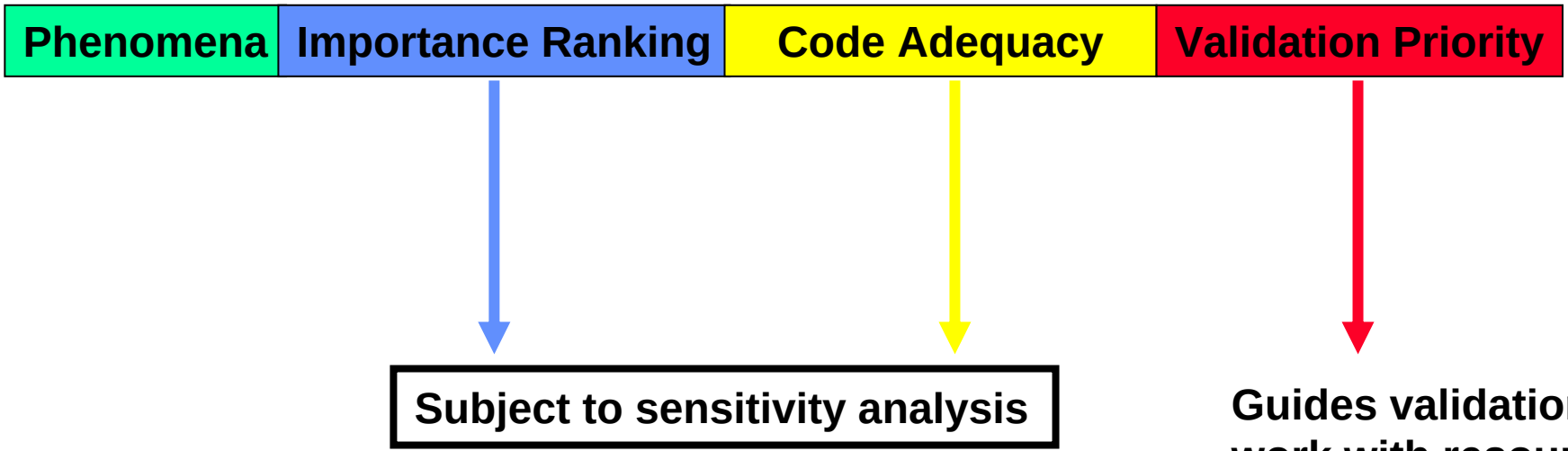
  Calibration is dangerous.

The real world:

There is a strong tendency to mix up verification, validation, and calibration that MUST BE RESISTED for high consequence computing.

The Right Experiments:

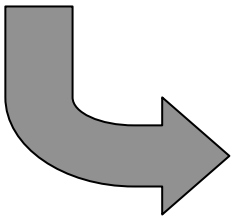
Phenomenology Identification and Ranking Table



The PIRT is subject to iteration because sensitivity analysis changes as work proceeds:

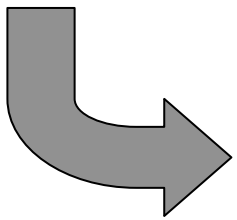
PIRT #1

Phenomena	Importance Ranking	Code Adequacy	Validation Priority



PIRT #2

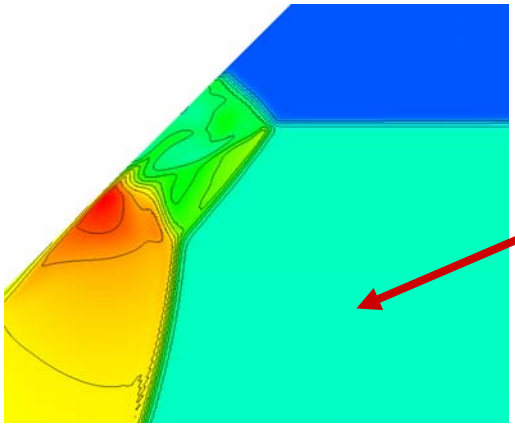
Phenomena	Importance Ranking	Code Adequacy	Validation Priority



PIRT #N

Phenomena	Importance Ranking	Code Adequacy	Validation Priority

The right calculations:



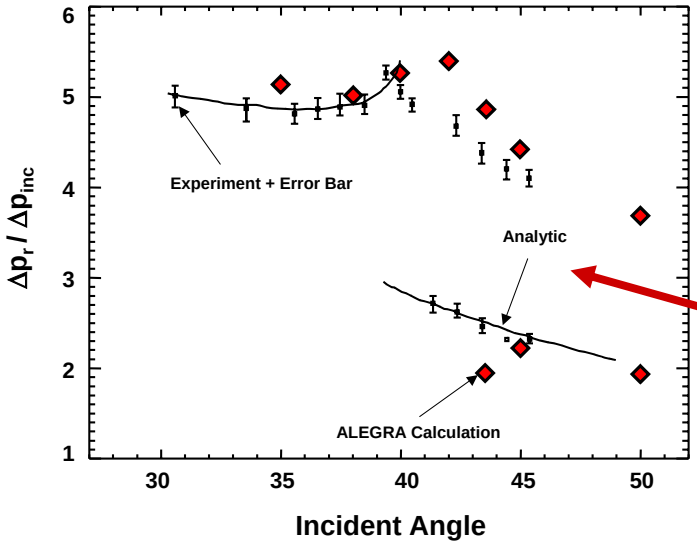
- Code bugs?
- Test what?
- Numerical performance (consistency, stability, convergence)?
- Numerical robustness?
- Calculations are sensitive to what?

The code: $M(\vec{p})$

- Multi-physics
- Multi-resolution
- p is a (large) parameter including parameters required to specify physics, numerics, scenarios
  - The parameter vector p is typically high-dimensional, especially if the grid specification is part of the parameter list
  - Verification centers on the numerics components of p.
- Verification must be and is prioritized by the PIRT (verify what you are trying to validate).
- Sensitivity analysis is required.

SAND2004-1505C

Comparing the right way: "Validation Metrics"



- Accurate calculations?
- Accurate experiments?
- Uncertainty accounted for in comparisons?
- Comparisons relevant?

- **Validation** compares code results $M(\vec{p}_1),..,M(\vec{p}_N)$ with experimental **benchmarks** $E(\vec{p}_1),..,E(\vec{p}_N)$ for a directed choice of $\vec{p}_i$
- Validation metrics quantify the difference, accounting for uncertainty.

$$D\left\{M(\vec{p}_i), E(\vec{p}_i)\right\}$$

- The parameters $\vec{p}_i$ vary over the physics, not over the numerics.
- It is often the case that $N \ll \dim(\vec{p})$, so sensitivity analysis is very important for best leveraging limited experiments.
- Note that a simple definition of prediction is now any $M(\vec{p})$ for which $\vec{p} \neq \vec{p}_i, i = 1,\ldots,N$; such values are usually inputs into important decisions.

The right conclusions:

- The goal of validation is to measure credibility of the code for an intended application:

$$C_{red}\left[D\left\{\boldsymbol{M}\left(\vec{p}_1\right),\boldsymbol{T}\left(\vec{p}_1\right)\right\},\ldots,D\left\{\boldsymbol{M}\left(\vec{p}_N\right),\boldsymbol{T}\left(\vec{p}_N\right)\right\}\right]$$

- This puts a premium on the quality of the validation metrics:
  - Converged calculations?
  - Guaranteed no code bugs?
  - Experimental uncertainty (variability and bias) quantified? Replicated in the calculations?
  - Experimental sensitivity matched by code?
- Decisions depend on our assessment of credibility.
- How sensitive are decisions to the various factors?

What is a credibility function?

- Examples appear in normal statistical software reliability theory:
  - For example, consider the number of "failures" in the time interval [0,t] $N(t)$
  - Assumptions lead to the description of $N(t)$ as a Poisson process, and allows the calculation of things like probability of k failures in [0,t], probability of a failure in [t,2t], probable time of k+1st failure, etc.
  - Credibility, for example, increases if probable time of next failure is large, or likely number of future failures is small.
- What is a "failure" for computational science? Probably an extension of reliability theory, such as:
  - A validation metric that is too large.
  - Too many failed experimental comparisons.

Return to calibration:

- **Credibility and calibration don't have to use the same formalism:**

$$C_{red}\left[D\left\{\boldsymbol{M}\left(\vec{p}_1\right),\boldsymbol{T}\left(\vec{p}_1\right)\right\},\ldots,D\left\{\boldsymbol{M}\left(\vec{p}_N\right),\boldsymbol{T}\left(\vec{p}_N\right)\right\}\right]$$

$$\min_{\hat{p}\subset\vec{p}\in\Omega} C_{al}\left[D_{cal}\left\{\boldsymbol{M}\left(\vec{p}_1\right),\boldsymbol{T}\left(\vec{p}_1\right)\right\},\ldots,D_{cal}\left\{\boldsymbol{M}\left(\vec{p}_N\right),\boldsymbol{T}\left(\vec{p}_N\right)\right\}\right]$$

- **Calibration should acknowledge credibility, hence what is known about the results of validation:**

$$\min_{\hat{p}\subset\vec{p}\in\Omega} C_{al}\left[D_{cal}\left\{\boldsymbol{M}\left(\vec{p}_1\right),\boldsymbol{T}\left(\vec{p}_1\right)\right\},\ldots,D_{cal}\left\{\boldsymbol{M}\left(\vec{p}_N\right),\boldsymbol{T}\left(\vec{p}_N\right)\right\};C_{red}\right]$$

- **We are currently investigating calibration formalisms accounting for model uncertainty, such as that due to Kennedy and O'Hagan or found in machine learning theory, with this goal in mind,.**

Calibration and Validation: Who Cares?

- Scientists care: about calibration and V&V, and their role in R&D
  - Center of gravity is scientific progress.
  - "It's so beautiful it has to be right!"*
- Code developers care: about V&V
  - Center of gravity is testing their software (users are testers).
  - "We built a really good code, but nobody used it!"*
- Decision makers care: about prediction
  - Center of gravity is spending money and risking lives.
  - "We scientists do the best we can; we can't be held legally liable for mistakes!"*
- Measures of success are not necessarily the same for these key groups.

*Quotes it's been my displeasure to hear over the past seven years.

Conclusions:

- Anything dealing with code calculations starts with verification.
- Validation and calibration are different.
- Disguising calibration as validation is dishonest.
- Calibration is dangerous in high-consequence computing (latest example is the use of CRATER – *AN ALGEBRAIC MODEL* – in the Columbia flight); the danger may be reduced by careful acknowledgement of the results of a rigorous validation effort during calibration.
- Prediction with a quantified basis for confidence remains the most important problem.

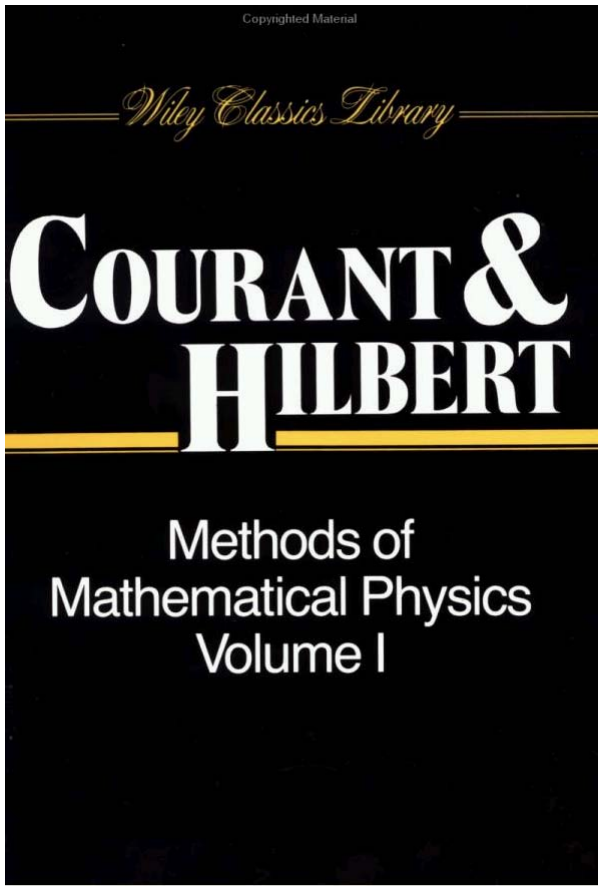
Conclusion:

"We make no warranties, express or implied, that the programs contained in this volume are FREE OF ERROR, or are consistent with any particular merchantability, or that they will meet your requirements for any particular application. THEY SHOULD NOT BE RELIED UPON FOR SOLVING A PROBLEM WHOSE SOLUTION COULD RESULT IN INJURY TO A PERSON OR LOSS OF PROPERTY…" [Emphasis Mine] (from Numerical Recipes in Fortran, Press, Teukolsky, Vetterling, and Flannery)

Will we be able to seriously claim that ASCI codes are any better than this?!

How absurd would the following be?

How much more absurd would the following be?



We make no warranties, express or implied, that the book you are about to read is free of error…